

## Criando a relação de Livro com Autor

### Transcrição

Nosso próximo objetivo será permitir o cadastro de um ou mais autores para os livros.

Vamos abrir nosso `form.xhtml`, e inserir mais um campo, com `h:selectManyListbox` que nos permitirá selecionar mais de um autor, como no exemplo abaixo:

```
<div>
  <h:outputLabel value="Autores" />
  <h:selectManyListbox>
    <f:selectItems value="#{adminLivrosBean.autores}"
      var="autor"
      itemValue="#{autor}"
      itemLabel="#{autor.nome}" />
  </h:selectManyListbox>
</div>
```

Observe os atributos informados na `f:selectItems`, dentro dele temos o atributo `value` que servirá para informar a lista de autores que vem do nosso servidor. Além disso, temos ainda mais dois atributos importantes, o `itemLabel` e também o `itemValue`. O `label` serve para dizer ao JSF qual será o valor exibido para o usuário selecionar, mas esse valor não serve para relacionarmos no nosso banco de dados, por isso temos o `itemValue`, onde diremos qual valor será colocado na nossa entidade. Para `value` e `label` usamos a expression language do JSF com o `#{auto...}`, mas de onde ele veio?

Nossa lista de autores, está no formato `List`, mas queremos apenas um único `Autor` dessa lista, assim usamos um outro atributo chamado `var`, que passa para cada elemento da lista, o valor dele para dentro do atributo `var` que será nosso autor.

Já que o `h:selectManyListbox` está recebendo como valor o `adminLivrosBean.autores`, temos que criar em nosso arquivo `AdminLivrosBean` um método chamado `getAutores` que devolve uma lista de `Autor`. Vamos criá-los.

```
public List<Autor> getAutores(){
}
```

Mas ainda não criamos a entidade `Autor`, vamos criá-la no pacote `br.com.casadocodigo.loja.models`. Nosso autor deverá parecer-se com:

```
public class Autor {

    private Integer id;

    private String nome;

    public Autor(Integer id, String nome) {
        this.id = id;
    }
}
```

```
        this.nome = nome;
    }

    // gere os getters e setters de id e nome

}
```

Voltando para a classe `AdminLivrosBean`, agora conseguimos pegar os autores e retornar para o xhtml.

```
public List<Autor> getAutores(){
    return Arrays.asList(new Autor(1, "Paulo Silveira"), new Autor(2, "Sérgio Lopes"));
}
```

Uma vez que temos nossa lista de autores pronta, já podemos subir nosso servidor para testar. Os autores ainda não estão vindo do banco de dados, porém vamos resolver isso logo em seguida.

